行政院國家科學委員會專題研究計畫成果報告

計畫編號：NSC 87-2213-E-009-049
執行期限：86 年 8 月 1 日至 87 年 7 月 31 日
主持人：陳正　　Email：cchen@eicpca5.csie.nctu.edu.tw
執行單位：國立交通大學資訊工程研究所

一、中文摘要

接續前兩年之研究內容，我們在第三年中完成了可延伸性多處理機記憶體子系統整合測試及評估，以得到可延伸性多處理機記憶體子系統之具體模擬評估結果。其重點包含了可延伸性多處理機記憶體子系統模擬評估境雛形之完成、對於 clustering-based MP 系統架構之整體考量。

關鍵詞：可延伸性、多處理機、叢集、快取記憶體一致性協定、記憶體一致性模組、連結網路

Abstract

Following the previous two years research, we have completed the implementation and evaluation of the scalable multiprocessor memory subsystem during the third year; including simulation and evaluation of scalable multiprocessor memory subsystem prototype and clustering-based Multiprocessor.

Keywords：Scalability、Multiprocessor、Cluster、Cache coherence protocol、Memory consistency model、Interconnection network

二、緣由與目的

由於電腦應用程式所需的計算量日益增加，使得平行處理機系統架構成為未來電腦系統發展的趨勢。在平行處理機(Parallel Processors)系統中，依其架構特性可分成分散記憶體多計算機(Distributed-memory Multicomputers)與共享記憶體多處理機(Shared-memory Multiprocessors)兩種不同的架構。而由於 Cost-effectiveness 和 Easy-programming 的考量，使得共享記憶體多處理機系統已成為近年來熱門的設計方式[1][2][3][4]。然而，在設計一個共享記憶體多處理機系統時，共享記憶體子系統設計將會明顯地影響整體系統效能的好壞與可延伸性(Scalability)。因此，如何設計一個良好之記憶體子系統以提昇整體系統效能，將成為共享記憶體多處理機系統設計上相當重要的一環。而且 VLSI 與 package 的技術不斷進步[5][6]，近年來在多處理機系統的架構中，逐漸有人嘗試以將數個處理器封裝到一個叢集(Cluster)，藉以利用記憶體存取上的區域性(Locality)來達到提昇系統效能的目的並探討此一架構對整體系統效能的衝擊，來找出未來設計多處理機系統時，具有可延伸性並且效能更高的架構。在叢集多處理機架構中，有兩個重要的考量方向：一是系統的可延伸性(Scalability)、一是系統架構是否能充分利用應用程式(Application) 中的溝通區域性(Communication Locality)[Basa 5]。

所以，在本計劃中，我們針對共享記憶體多處理機系統設計上重要的議題(Issues)，進行模擬評估與分析，以作為設計記憶體子系統的重要參考資料。另外，在本論文中我們將設計一個叢集多處

理機系統，希望能夠發揮叢集多處理機系統的可延伸性與區域性，來得到一個效果良好的多處理機系統。透過模擬評估的結果，我們將瞭解到，在設計叢集多處理機系統時的一些重要考量因素，如叢集系統的可延伸性與區域性等優點、叢集大小以及叢集間快取記憶體大小應如何調整其設定等。

四、計劃結果自評

在可延伸性多處理機記憶體子系統模擬評估境雛形的部份，我們完成了下面這些架構：

(1)Directory-based cache coherence protocol
- SCI
- write-invalidate
- write-update
- competitive-update
- clean

(2)Relaxed Memory Consistency Models：
- sequential consistency model
- processor consistency model
- weak consistency model
- release consistency

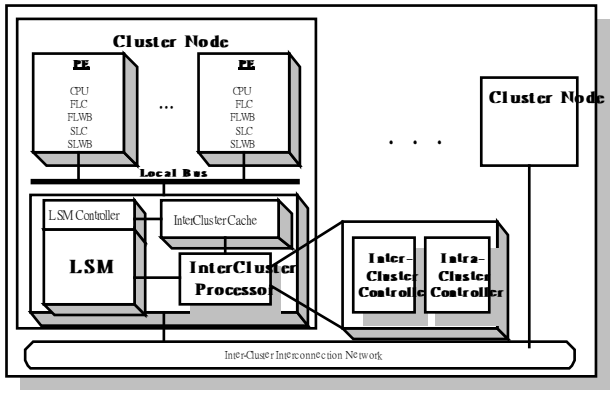
(3)Bus 架構
- split-trasaction bus
- simple bus

(4)Interconnection Network
- K-ary, N-cube topology
- 傳送的頻寬
- 傳送的方向性
- 傳送時的 wire delay　switch delay
- flit 大小

(5)其它
- Two-level cache
- wire buffer
- wire cache
- 抱括對 migratory-sharing access 及 synchronization 等最佳方式

(6)系統架構
- 非叢集架構
- 叢集架構
- PMP 架構

另外，在 clustering-based MP 系統架構之整體考量部份，我們也設計出了我們自己的叢集多處理機記憶體子系統，如圖一所示，由圖中可以看出本架構圖中，一個 Cluster Node 包含下列各元件：



圖一 我們所設計的叢集分散式共享記憶體多處理機系統架構圖

(1)數個 Processor Environments (PE)，每個 PE 由 CPU、FLC、FLWB、SLC、SLWB 所組成，其概略圖如圖 2 所示。因為本論文旨在探討叢集記憶體子系統，所以有關處理機架構，我們採用結構簡單的 RISC 之處理機架構且為具有 Blocking-Load 的特性[7]。利用 Two-Level Cache 來提高 Local 的 Hit ratio 則可縮短記憶體存取的延遲時間。由於我們採用釋放記憶體一致性模組[8]來當我們的記憶體一致性模組，故利用 Two-Level Write buffer 來提供記憶體存取更多 Buffering 的能力，以提供放

鬆的記憶體一致性模組的模擬方式。

(2)一個 Local Shared Memory(LSM)，所以我們的系統屬於分散式快取記憶體。將共享記憶體分散到每一個叢集節點(Cluster Node)中，此可降低記憶體衝撞(Contention)以及提高 Data locality。

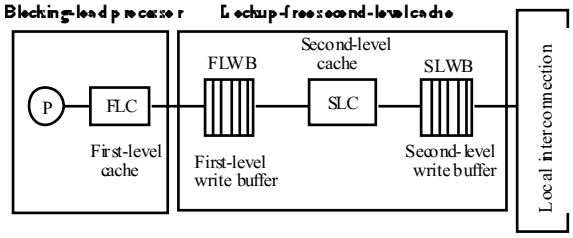


圖 2 Processor Environment 的架構概略圖

(3)一個叢集間快取記憶體(Inter Cluster Cache)，每個叢集有自己的 Inter-Cluster Cache，功能和前面二層快取記憶體相同，是為了要提供叢集內的處理器另外一個分享的快取記憶體區間，以充分利用資料的區域性。然而，叢集間快取記憶體當成是一個叢集內數個 PE 的第三層快取記憶體。

(4)一條匯流排(Local Bus)，為叢集內各處理器、叢集間快取記憶體與 Local Shared Memory 的 intra-connection network。利用 Bus 當我們的叢集內 PE(Processor Enviornment)的溝通管道，可以加快叢集內 PE 之間的訊息傳輸速度。我們所模擬的 Bus 為 split bus，可以防止延遲長的存取占用 Bus 太多的時間。

(5)一個 Inter Cluster Processor (ICP)，內部包含了輸入輸出控制器(I/O Controller)。其功能為 Cluster Node 發出或接收 request 和 ack 封包(packet)，以及接收(並再發出)其它叢集節點發出且當未到達目的地的封包的界面。

(6)另外，Cluster Node 之間則是用可延伸的 k-ary n-Cube bus-c[basa 93]靜態網路結構，此架構可解決 Bus 結構下延伸性不足的問題。

並且透過模擬評估的結果，我們得到了設計叢集多處理機系統時的一些重要考量因素，如叢集系統的可延伸性與區域性等優點、叢集大小以及叢集間快取記憶體大小等。

最後，值得一提的是我們所發展出來的模擬評估環境 SEECMA(A Simulation Evaluation Environment for Cluster-based Multiprocessor Architecture)具有下列特色：

(1)模組化設計：我們的 SEECMA 對叢集系統設計者而言可以說是一個非常好的叢集環境雛形。因為它具有模組化設計的持性，使得使用者能夠很容易地針對其它的模擬項目加以擴充。

(2)多組模擬參數選項：我們建立的 SEECMA 提供多項模擬參數選項，讓使用者能夠透過這些選項，作不同的評估和探討。因此，它可以說是叢集多處理機的設計者在實作前的評估平台，及平行編譯技術研究者的測試平台。

(3)圖形化介面：為了配合使用者在操作上的方便，SEECMA 提供有 solaris 作業系統上的圖形化介面，可以讓使用者在設定時更快速、更方便。圖形化介面之使用範例請參閱附錄四。

五、參考文獻

[1] D. Lenowshi, and J. London, *Stanford DASH Multiprocessor*, Technical Report No. CS-TR-89-403, Dec 1989

[2] **Lenoski. D. E., et al. "*The Stanford DASH multiprocessor*", IEEE Comput. Vol.25, No.3, pp.63-79, Mar,1993**

[3] **Agarwal, A., et al. The MIT Alewife Machine:"*A large-scale distributed-memory multiprocessor*".In Dubois, M., and Thakkar, S. S.(Eds.).Scalable Shared Memory Multiprocessors. Kluwer, Dordrecht, pp.239-261,1991.**

[4] **Kendall Square Research. Technical Summary of the KSR1.1992。**

[5] Debashis Basak and Dhabaleswar K. Panda . *Scalable Architectures with k-ary n-cube cluster-c Organization.* TR28-1993, Dept. of Computer and Information Science, The Ohio State University, Aug 1993.

[6] Debashis Basak and Dhabaleswar K. Panda. *Designing Processor-cluster Based Systems: Interplay Between Cluster Organizations and Collective Communication Algorithm.* OSU-CISRC-1/96-TR05, Dept. of Computer and Information Science, The Ohio State University, 1996.

[7] H. Nilsson and P. Stenstrom, "*An adaptive update-based cache coherence protocol for reduction of miss rate and traffic*", In: Proc. PARLE Conf., Athens, Greece, pp. 336-374. July 1994.

[8] Kourosh Gharachorloo，Daniel Lenoski，James Laudon，Phillip Gibbons，Anoop Gupta，and John Hennessy. "*Memory Consistency and Event Ordering in Scalable Shared-Memory Multiprocessors*", In : Proc. 17th Annual International Symposium on Computer Architecture, pp. 15-26, May 1990.